Computational Complexity and Feasibility of Data Processing and Interval Computations. By Vladik Kreinovich, Anatoly Lakeyev, Jiří Rohn and Patrick Kahl. Kluwer Academic Publishers, Dordrecht. (1998). 459 pages. $214.00, NLG 375.00, GBP 128.00.

Contents:

Preface. 1. Informal introduction: Data processing, interval computations, and computational complexity. 2. The notions of feasibility and NP-hardness: Brief introduction. 3. In the general case, the basic problem of interval computations is intractable. 4. Basic problem of interval computations for polynomials of a fixed number of variables. 5. basic problem of interval computations for polynomials of fixed order. 6. Basic problem of interval computations for polynomials with bounded coefficients. 7. Fixed data processing algorithms, varying data: Still NP-hard. 8. Fixed data, varying data processing algorithms: Still intractable. 9. What if we only allow some arithmetic operations in data processing? 10. For fractionally-linear functions, a feasible algorithm solves the basic problem of interval computations. 11. Solving interval linear systems is NP-hard. 12. Interval linear systems: Search for feasible classes. 13. Physical corollary: Prediction is not always possible, even for linear systems with known dynamics. 14. Engineering corollary: Signal processing is NP-hard. 15. Bright sides of NP-hardness of interval computations I: NP-hard means that good interval heuristics can solve other hard problems. 16. If input intervals are narrow enough, then interval computations are almost always easy. 17. Optimization—A first example of a numerical problem in which interval methods are used: Computational complexity and feasibility. 18. Solving systems of equations. 19. Approximations of interval functions. 20. Solving differential equations. 21. Properties of interval matrices I: Main results. 22. Properties of interval matrices II: Proofs and auxiliary results. 23. Non-interval uncertainty I: Ellipsoid uncertainty and its generalizations. 24. Non-interal uncertainty II: Multi-intervals and their generalizations. 25. What if quantities are discrete? 26. Error estimation for indirect measurements: Interval computation problem is (slightly) harder than a similar probabilistic computational problem. Appendices. A. In case of interval (or more general) uncertainty, no algorithm can choose the simplest representative. B. Error estimation for indirect measurements: Case of approximately known functions. C. From interval computations to modal mathematics. D. Beyond NP: Two roots good, one root better. E. Does "NP-hard" really mean 'intractable"? F. Bright sides of NP-hardness of interval computations II: Freedom of will? G. The worse, the better: Paradoxical computational complexity of interval computations and data processing. References. Index.

The Patterns Handbook: Techniques, Strategies, and Applications. Edited by Linda Rising. Cambridge University Press, New York. (1998). 549 pages. $44.95.

Contents:

About the editor. Foreword (James Coplien). Preface. Acknowledgments. I. Overview. Design patterns: Elements of reusable architectures (Linda Rising). An overview of patterns (Russell Corfman). Patterns: Spreading the word (Linda Rising). A training experience with patterns (Brandon Goldfedder and Linda Rising). Patterns: The new building blocks for reusable software architectures (Diane Saunders). II. Examples and experience. Experience in applying design patterns to decouple object interactions in the INgage^{TM} IP prototype (Michael Duell). Pattern writing (Linda Rising). Writers workshop format. AGCS pattern template. Patterns mining (David E. DeLano). System test pattrn language (David E. DeLano and Linda Rising). Improving software development with process and organizational patterns (Patricia Genauldi). Organizational patterns at AG Communication Systems (Norm Janoff). HandsInView (Don Olson). Patterns on the fly (Don S. Olson). A pocket-sized broker (Don S. Olson). Frameworks and design patterns (Ben H. Richards). III. Resources and more information. Fault-tolerant telecommunication system patterns (Michael Adams, James Coplien, Robert Gamoke, Robert Hanmer, Fred Keeve and Keith Nicodemus). Industrial experience with design patterns (Kent Beck, James O. Coplien, Ron Crocker, Lutz Dominick, Gerard Meszaros, Frances Paulisch and John Vlissides). Sorting through the plethora: The "unofficial" JOOP Book Awards (Steven Bilow). Patterns (Grady Booch). A generative development—Process pattern language (James O. Coplien). Setting the stage (James O. Coplien). Software design patterns: Common questions and answers (James O. Coplien). Software development as science, art, and engineering (James O. Coplien). The failure of pattern languages (Richard P. Gabriel). Potential pattern pitfalls, or How to jump on the patterns bandwagon without the wheels coming off (Neil B. Harrison). An introduction to patterns (Ralph E. Johnson). How patterns work in teams (Ralph E. Johnson). A report on PLoP'94 (Ralph E. Johnson). Patterns and frameworks (Ralph E. Johnson). Patterns and antipatterns (Andrew Koenig). Design reuse: Chemical engineering vs. software engineering (Paul Kogut). Christopher Alexander: An introduction for object-oriented designers (Doug Lea). Patterns: PLoP, PLoP, fizz, fizz (Robert Martin). A design patterns experience report (Russell L. Ramirez). Design patterns to construct the hot spots of a manufacturing framework (Hans Albrecht Schmid). Using design patterns to evolve system software from UNIX to Windows NT (Douglas C. Schmidt and Paul Stephenson). Pattern hatching—Perspectives from the "Gang of Four" (John Vlissides). Appendices. A. Annotated bibliography. B. Web sites. Index.

Introduction to the Theory of Error-Correcting Codes (Third edition). By Vera Pless. John Wiley & Sons, New York. (1998). 207 pages. $59.95.

Contents:

Preface. 1. Introductory concepts. 2. Useful background. 3. A double-error-correcting BCH code and a finite field of 16 elements. 4. Finite fields. 5. Cyclic codes. 6. Group of a code and quadratic residue (QR) codes. 7. Bose-Chaudhuri-Hocquenghem (BCH) codes. 8. Weight distributions. 9. Designs and games. 10. Some codes are unique. Appendix. References. Index.